REMARKS

1. INTRODUCTION

Applicant has amended claim 13. Accordingly, claims 1 - 36 remain pending and under consideration in this application. Reconsideration and reexamination is hereby respectfully requested.

2. CLAIM REJECTION UNDER 35 U.S.C. § 101

Claim 13 stands rejected under 35 U.S.C. § 101. In this regard the Office Action states that the recitation "performance of said method" constitutes an abstract idea. Applicant respectfully submits that the rejection has been overcome through amendment.

Claims 16-25 stand rejected, and in this regard the Office Action states that the claimed invention is directed to non-statutory subject matter. Applicant respectfully traverses the rejection.

Applicant respectfully directs the Examiner's attention to § 2106 of the MPEP *Patentable Subject Matter – Computer-Related Inventions*:

> Computer programs are often recited as part of a claim. Office personnel should determine whether the computer program is being claimed as part of an otherwise statutory manufacture or machine. In such a case, the claims remains statutory irrespective of the fact that a computer program is included in the claim. The same result occurs when a computer program is used in a computerized process where the computer executes the instructions set forth in the computer program. Only when the invention taken as a whole is directed to a mere program listing, *i.e.*, to only its description, is it descriptive material per se and hence non-statutory.

Section IV(B)(1)(a), MPEP § 2106.

Independent claim 16 recites "A method of detecting memory leaks for a program executing on a computer . . .." Applicant respectfully submits that the claim as a whole does not attempt to claim a program listing—the descriptive material. Rather, Claims 16-25 are directed to a statutory process, which statutory process ("A method of detecting memory leaks . . .")

operates on the "program executing on a computer." Accordingly, pursuant to the MPEP, "only when the claimed invention taken as a whole is directed to a mere program listing, *i.e.*, to only its description or expression, is it descriptive material *per se* and hence nonstatutory." Applicant respectfully requests reconsideration and withdrawal of the rejection of claims 16-25.

3. CLAIM REJECTION UNDER 35 U.S.C. § 102

Claims 1-36 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. 2004/0078540 (Cirne et al.). Applicant respectfully overcomes this rejection.

Cirne et al. Do Not Monitor Allocated Memory Levels

As an initial matter, the Office has read the various recitations of "allocated memory levels" in the claims on the disclosures (in Cirne et al.) involving tracking growth patterns of groups of stored items (*e.g.*, one example is a Java collection). Applicant wishes to point out that while in general, an increase in the number of objects will correspond with an increase in memory consumption, this is not necessarily always true since there is not a direct relationship between objects and memory. In most implementations of object management, a block of memory is allocated and may be used to store one object, multiple objects, or a variable number of objects. Accordingly, the claimed inventions involving "allocated memory levels" (as opposed to objects) are not only different and not disclosed by Cirne et al, but are rather more complicated as well. This complexity arises because the allocation of objects in memory blocks creates an additional layer of abstraction. The correlation between objects and memory worsens significantly beyond the simple situation of one-object-per-memory block, for example, in the situations where there are multiple objects per memory block, the number of objects per block are growing, or the number of objects per memory block are variable. Changes in the size (i.e., number) of objects do not necessarily alter allocated memory.

In sum, Applicant traverses the rejection of all claims 1-36 that are based on Cirne et al. since Cirne et al. at most teach tracking the size of a group (of objects)—thus an object leak methodology, and not a method and system of memory leak detection as claimed.

Moreover, while the Office has not made an obviousness rejection, Applicant, and with complete reservation of rights to address a specific obviousness rejection should it be made in the future, submit that the present invention is non-obvious in view of Cirne et al. due to the distinctions between objects and allocated memory set forth above.

Even assuming for the purpose of response only that "allocated memory" can be read on "object size" (which it cannot—see above), the claimed inventions nonetheless define novel and nonobvious subject matter, as explained in the remainder of this response, and accordingly should be allowed.

Peak Allocated Memory Level

Independent claims 1, 16 and 26 each recite "when a peak allocated memory level has increased a determined number of times." Cirne et al. do not teach or suggest this recitation.

Cirne et al. disclose technology that purports to identify potential sources of memory leaks by tracking growth patterns of groups of stored items. Paragraph [0015]. "If the growth pattern of a collection indicates that it may be the source of a memory leak, that collection is reported to a user . . .." [0015]. Cirne et al., however, do not provide a definition of what constitutes a "growth pattern." Applicant has carefully reviewed Cirne et al. and find only one discrete example of what is meant by "growth pattern." In paragraph [0045] Cirne et al. discloses that "[a]gent 8 looks for collection instances that appear to be constantly growing in size (*i.e.*, the number of objects stored in the collection grows)." Thus, even assuming for the purposes of argument only that growth in the number objects stored corresponds to a growth in the memory used (see above), Cirne et al. at most only disclose a growth pattern that is ***constantly growing in size.*** This approach does not meet the limitation of determining a "peak allocated memory level" and further determining when that peak allocated memory level has increased a determined number of times, as positively claimed.

For example, the teaching of Cirne et al., namely, a growth pattern that is "constantly growing in size" *excludes* the very example Applicant provided in its own Figure 3, particularly between peak P_4 (time t_5) and peak P_5 (time t_6). Thus, while Applicant's process of monitoring for increases in the "peak allocated memory levels" will identify Figure 3, Cirne et al.'s "constantly growing in size" methodology will fail to identify Applicant's Figure 3 as involving a memory leak because the allocated memory level is not "constantly growing" between t_5 and t_6 but rather includes a decrease in between. In sum, Cirne et al. simply do not teach identifying memory leaks based on the specific approach involving "peak allocated memory levels."

Accordingly, for at least this reason, Applicant respectfully requests reconsideration and withdrawal of the rejection of independent claims 1, 16 and 26.

Moreover, claims 2-15, 17-25 and 27-36 depend from claims 1, 16 and 26, respectively, and therefore include all of the limitations thereof. Accordingly, for at least the same reasons, Applicant respectfully requests reconsideration and withdrawal of the rejection of these claims.

Filtering Out Increases in Peak Allocated Memory Levels Not Indicative of a Memory Leak

Moreover, independent claims 16 and 26 each recite "filtering out increases in peak allocated memory levels not indicative of a memory leak." Cirne et al. does not teach or suggest this recitation.

First, Cirne et al. make no disclosure whatsoever of identifying, tracking and using a "peak allocated memory level" for any purpose, much less applying any filtering function to those peak levels where such peaks are not indicative of a memory leak. In this regard, the Office Action refers to paragraph [0017] of Cirne et al. for the proposition that purports to meet this limitation: "reporting a flagged collection which no longer appears to be leaking." Applicant believes the Office is actually referring to paragraph [0016] for this proposition, which is reproduced below: "If a flagged collection no longer appears to be leaking, that change in status will be reported . . .."

In any event, Cirne et al. provide NO explanation whatsoever as to the underlying logic that caused its system to conclude that a collection is no longer leaking and certainly no explanation that the change in status from "LEAKING" to "NOT LEAKING" is the result of filtering out increases in peak allocated memory levels. This passage in Cirne et al. describes a simple reporting function. Accordingly, Applicant respectfully submits that Cirne et al. does not teach or suggest filtering increases in peak allocated memory levels not indicative of memory leak. This function is important since increases in the peak allocated memory levels *not* indicative of memory leak--that might otherwise trigger a memory leak condition--can be suppressed (*i.e.*, helpful in minimizing false positives).

Accordingly, for at least this reason, Applicant respectfully requests reconsideration and withdrawal of the rejection of independent claims 16 and 26.

Additionally, claims 17-25 and 27-36 depend from claims 16 and 26, respectively, and therefore include all of the limitations thereof. Accordingly, for at least this additional reason, Applicant respectfully requests reconsideration and withdrawal of the rejection of these claims.

Moreover, dependent claims 3 and 14 further include this recitation and is allowable for the same reason.

Ignoring Increases In Peak Allocated Memory Levels During A Startup Time Interval

Claims 4, 14 and 28 positively recite that the "filtering step includes the substeps of: ignoring increases in peak allocated memory levels during a startup time interval immediately after said program begins to execute . . . ." Cirne et al. do not teach or suggest this feature. In this regard, however, the Office has cited to paragraphs [0016] – [0017] of Cirne et al. for the proposition that purports to satisfy this feature, namely, "discontinuing track of newly allocated collections if no longer appear to be leaking." Applicant disagrees.

Applicant has carefully reviewed the cited paragraphs, and takes note of the disclosed time-out period. Applicant, however, points out that while this is a similar term (the claimed "startup time" versus the disclosed "time-out period") the function is completely different, and in-fact is OPPOSITE. Cirne et al. ONLY collects data during its "time-out period." On the contrary, however, according to the invention, data collection is EXCLUDED during its "startup time." This has the advantage in the present invention of disregarding potentially misleading memory allocations that occur when a program begins initial execution. In Cirne et al., on the other hand, the purpose of limiting downstream data collection after the "time-out period" is to reduce the overhead burden of the system (see [0017]). In Cirne et al., if a monitored program does not immediately show a memory leak at the beginning of execution during its "time-out period" it will fall off the monitored list. This is a significant shortcoming of Cirne et al. relative to the present invention.

Therefore, for at least these additional reasons, Applicant believes claims 4, 14 and 28 are allowable and respectfully requests reconsideration and withdrawal of the rejection.

Ignoring Increases In Peak Allocated Memory Levels That Occur After Said Startup Time Interval But That Occur Less Than A Preselected Time Apart

Claims 5, 14 and 29 further recite that the "filtering step includes . . . ignoring increases in peak allocated memory levels that occur after said startup time interval but that occur less than a preselected time apart . . .." Cirne et al. do not teach or suggest this recitation. In this regard, however, the Office has cited to paragraph [0019] for the proposition that purports to satisfy this feature, namely, "reclassifying stored group which appears not to be leaking." Applicant disagrees.

Applicant has carefully reviewed the cited paragraph, and takes note of the disclosure made in Cirne et al. of a comparison between the current size of the group of objects being monitored and a threshold, in order to reclassify the status of that group from "leaking" to "not leaking." However, Applicant find nothing in [0019] or anywhere else in Cirne et al. that deals with determining the time interval between peaks in allocated memory levels (as claimed, the "preselected time apart"), much less the function of ignoring an increase in a peak allocated memory level *when* it is less than a "preselected time apart" (*i.e.*, measured relative to the previous "peak" allocated memory level). This notion of a "peak-to-peak" timer is not taught or suggested in Cirne et al. This feature is important since it allows the method and system of the present invention to treat many increases in the peak allocated memory levels that are closely-spaced in time as a single event for leak detection purposes. This has a significant advantage in eliminating false alarms.

Therefore, for at least these additional reasons, Applicant believes claims 5, 14 and 29 are allowable and respectfully requests reconsideration and withdrawal of the rejection.

Determining a Memory Leakage Rate

Independent claim 16 recites "determining a memory leakage rate . . .." Cirne et al. do not teach or suggest this limitation. In this regard, however, the Office cites to paragraphs [0016] and [0019] for the proposition that purports to satisfy this feature, namely, "tracking collection as metric data and reporting the tracking of group of stored items reported as being a potential source of a memory leak if the received size satisfies the current value of a threshold." Applicant disagrees.

Applicant has carefully reviewed the cited paragraphs and takes note that Cirne et al. at most disclose only a comparison between a collection size and threshold (size) (*i.e.*, is the collection size ***greater than*** the threshold). The result of the comparison is a YES or a NO. In-fact, this "collection size" is measured in terms of a number "objects" and may not necessarily even correspond to allocated memory levels. Notwithstanding this, in sum, Cirne et al. nonetheless make no disclosure of the relative amount by which the collection size exceeds the threshold, much less keep track of the series of such relative amounts as would be needed to determine a "memory leakage rate" (*e.g.*, X kilobytes per minute, as specified in paragraph [0030] of Applicant's published application under "8. MINIMUM_RATE_FOR_ALARM"). Cirne et al. cannot satisfy this limitation.

Dependent claims 6-7, 15, 30-31 and 36 all include the recitation "memory leakage rate" which is respectfully submitted to lend an additional basis for patentability for the same reasons given above in connection with claim 16.

In view of the foregoing, Applicant respectfully submits that the rejection of all of the claims has been overcome, and respectfully requests reconsideration and withdrawal of the same.

4. CONCLUSION

For the foregoing reasons, all presently pending claims are now believed to be in a condition for allowance. Early notice of the same is hereby respectfully requested.

Respectfully submitted,

Date: June 9, 2006

By: ____________________
John W. Rees, Reg. No. 38,278
Dykema Gossett PLLC
Customer No. 26127
39577 Woodward Avenue, Suite 300
Bloomfield Hills, MI 48304
(248) 203-0832
ipmail@dykema.com

BH01\634193.2
ID\JWR